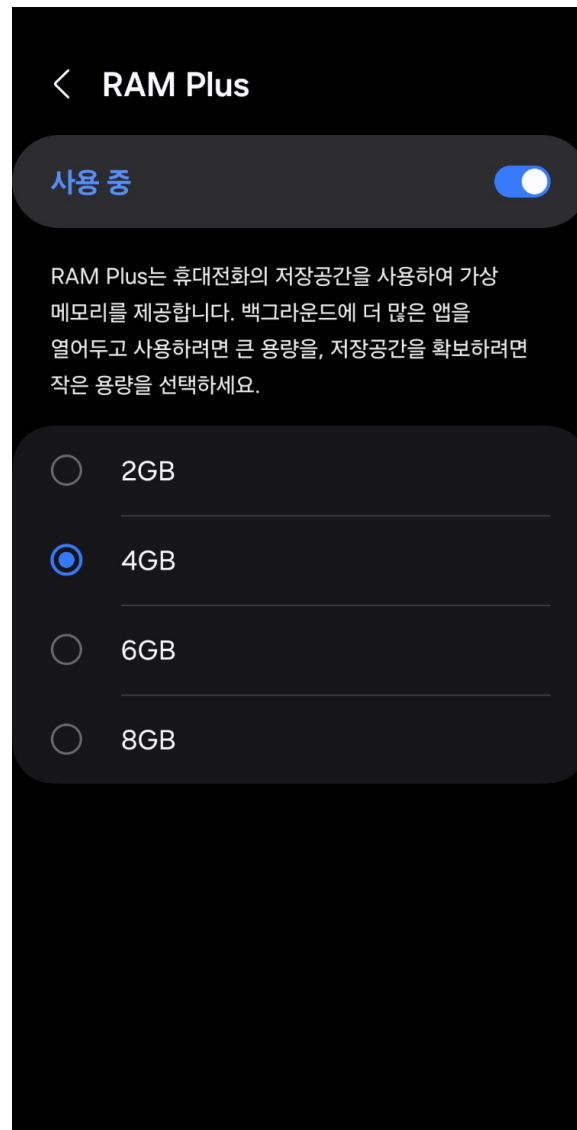


Virtual Memory

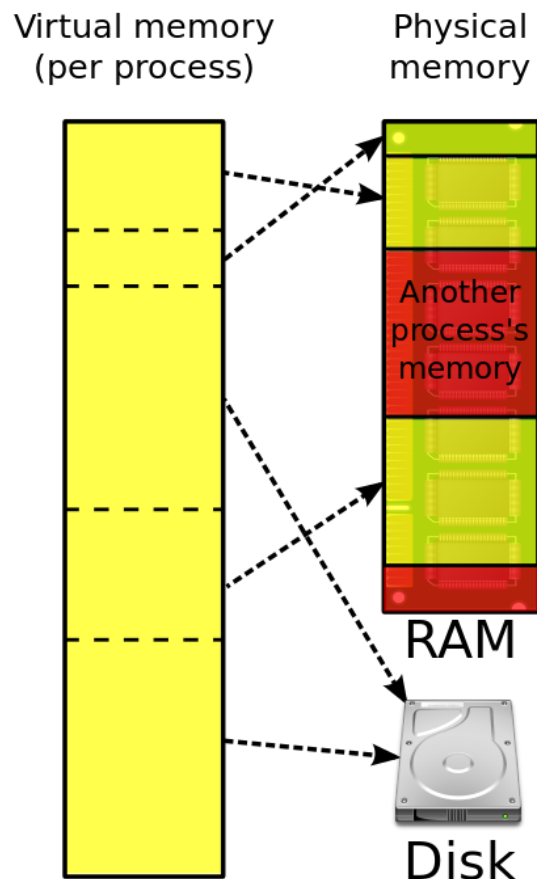
개요



가상 메모리의 개념은 1957년에 발표. 실제 적용된 것은 맨체스터 대학교가 Atlas용으로 1961년에 개발한 것이 최초임. 1965년에 MIT가 개발한 멀틱스 시스템 이후 본격적으로 채용되기 시작함.

가상 메모리 기법

메인 메모리의 영역을 하드 디스크까지 확장하여 사용하는 메모리 관리 기법. 컴퓨터 시스템 실제로 이용 가능한 기억 자원을 추상화하여 사용자에게 더 큰 용량의 메모리로 보이도록 만듦. 보통 OS에서 지원해줌.



가상 주소, 논리주소: 가상적으로 주어진 주소

물리 주소, 실제 주소: 실제 메모리 상에서 유효한 주소

가상 주소의 범위를 가상 주소 공간, 물리 주소의 범위를 물리 주소 공간이라고 함.

메모리 관리 유닛(MMU, Memory Management Unit)

CPU는 가상 메모리를 다루고, 가상 메모리의 가상 주소에 접근 시 MMU를 통해 물리 주소로 변환되어 물리 메모리에 접근한다.



페이지(Page)

가상 주소 공간을 분할한 일정한 크기의 블록. 페이지의 크기는 하드웨어에 의해 정의됨.

프레임(Frame)

물리 메모리를 일정한 크기로 나눈 블록.

가상 주소 공간을 페이지 단위로 나누고 실제 기억 공간은 페이지 크기와 같은 프레임으로 나누어 사용. 페이지가 하나의 프레임을 할당 받으면, 물리 메모리에 위치하게 됨. 프레임을 할당 받지 못한 페이지들은 외부 저장장치에 저장되며, 이 때도 프레임과 같은 크기 단위로 관리됨.

운영체제는 이러한 페이지들을 페이지 테이블 안에 집어 넣어서 관리.

프로세스가 가진 페이지 테이블 → **유저 메모리**

운영체제가 가진 페이지 테이블 → **커널 메모리**

프로세스는 각자 독립된 가상 주소를 가진다. 두 프로세스가 각자 0x1000라는 주소에 접근할 경우 CPU는 프로세스(컨텍스트)마다 가지는 페이지 테이블을 참조.

페이징(Paging)

프로세스의 가상 주소 공간과 실제 물리 주소를 일정한 크기의 페이지와 프레임으로 잘라서 이용하는 방법. 페이지들을 디스크에 적재하고 메인 메모리 가져와서 쓰는 게 페이지징.

페이지 테이블(Page Table)

가상 주소와 물리 메모리의 실제 주소를 매핑해놓은 테이블. MMU가 디스크에서 메인 메모리로 옮기려고 할 때 참조하는 테이블.

예시) A 프로세스의 페이지 테이블에서 0x1000 주소에 대응되는 실제 주소는 0x30000이고 B 프로세스의 페이지 테이블에서는 0x1000 주소에 대응되는 실제 주소가 0x40000일 때, 두 프로세스는 동일한 메모리 주소를 참조했지만 MMU가 프로세스의 페이지 테이블을 참조하여 실제 물리 메모리 주소를 얻고 그 위치를 참조해서 읽기 쓰기를 한 것. 따라서 두 프로세스의 주소 공간이 겹치지 않고 독립적으로 실행될 수 있으며 프로그래머는 다른 프로세스의 주소 공간을 생각할 필요가 없어짐.

요구 페이지징(Demand Paging)

프로세스를 구성하는 페이지 중 당장 필요한 페이지들만 메인 메모리에 적재하고 사용하지 않는 페이지는 디스크에 보관함.

- 메모리 사용량 감소.
- 프로그램에 필요한 물리 메모리 용량 감소



가상 메모리 기법에서 프로그램에 필요한 페이지가 보조기억장치에 있다면 해당 페이지를 메인 메모리로 가져올 때 성능 저하가 발생할 거 같다.

페이지 폴트(Page Fault)

페이지 테이블에는 유효 비트라는 게 존재함. 유효 비트를 통해 페이지가 실제 물리 메모리에 존재하는지 알 수 있음. 프로세스가 가상 주소에 접근하게 되어 MMU가 페이지 테이블을

참조한 뒤 페이지를 확인함. 이때 페이지가 유효하지 않을 경우 특정 트립을 발생시키는데, 그게 바로 **페이지 폴트**.

페이지 폴트를 감지한 운영체제는 CPU의 동작을 잠시 중단시킨 뒤 해당 페이지를 페이지징 파일에서 가져와서 물리 메모리의 비어있는 공간에 적재시키고 페이지 테이블을 최신화시킨 뒤 CPU의 동작을 재개시킴.

따라서 사용되지 않는 페이지를 페이지징 파일로 옮기는 것과 물리 메모리에 없는 페이지를 페이지징 파일에서 가져와서 물리 메모리에 적재하는 일은 운영체제가 자동으로 수행. 다만 **페이지 폴트는 그 과정 상 디스크에 접근하기 때문에 파일 입출력이 발생하므로 성능이 하락됨.**